Doc Code: AP.PRE.REQ

PTO/SB/33 (07-05)
Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Docket Number (Optional): 50277-2415

Pursuant to 37 CFR 1.8(a)(1)(ii) I hereby certify that this correspondence is being transmitted to the United States Patent and Trademark Office via the electronic filing system in accordance with 37 CFR §§1.6(1)(4) and 1.8(a)(1)(i)(C) on the date indicated below and before 9:00 PM PST.

on 07/09/09

Signature /AdamCStone#60531/

Typed or printed name Adam C. Stone

Application Number	Filed
10/815,220	03/30/2004
First Named Inventor	
Aravind Yalamanchi	
Art Unit	**Examiner**
2162	Stevens, Robert

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed.
(Form PTO/SB/96)

☒ attorney or agent of record.
Registration number 60,531.

☐ attorney or agent acting under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34

/AdamCStone#60531/
Signature

Adam C. Stone
Typed or printed name

(408) 414-1231
Telephone number

07/09/2009
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required.
Submit multiple forms if more than one signature is required, see below*.

☒ *Total of 1 forms are submitted.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Aravind Yalamanchi

Examiner: Robert Stevens

Art Unit: 2162

Serial No.: 10/815,220

Confirmation No.: 7098

Filed on: March 30, 2004

For: MANAGING EVENT-CONDITION-ACTION RULES IN A DATABASE SYSTEM

Sir:

PRE-APPEAL REQUEST FOR REVIEW ON BEHALF OF YALAMANCHI

This is a brief in support of a pre-appeal request for review of the Final Rejection mailed April 13, 2009 and the Advisory Action mailed June 25, 2009, in which currently-pending claims 42-44, 47-53, and 56-61 stand finally rejected. Applicant filed a Notice of Appeal and a Pre-Appeal Request For Review concurrently with this brief. This brief is submitted electronically in support of Applicant's pre-appeal request for review.

GROUNDS OF REJECTION TO BE REVIEWED

Whether independent claims 42 and 51 are unpatentable under 35 U.S.C. § 103(a) over S. Chakravarthy et al "Composite Events for Active Databases: Semantics, Contexts and Detection", Proc. of the 20th VLDB Conf, Santiago, Chile, 1994, pp. 606-617, pp. 42-48 (hereinafter "Chakravarthy") in view of Etzion et al U.S. Patent No. 6,604,093 (hereinafter "Etzion") and Hellerstein et al U.S. Patent Pub. No. 2002/0165842 (hereinafter "Hellerstein").

CLEAR ERROR IN REJECTION OF CLAIMS 42 AND 51

The references cited by the Examiner do not provide sufficient factual support for the rejection of claims 42 and 51 and therefore the rejection constitutes clear error. The Final Office Action likens Applicant's invention of claim 42 to Chakravarthy's composite event detector, but acknowledges that Chakravarthy and Etzion do not include any teaching of the following highlighted portions of Applicant's claim 42:

42. A computer-implemented method for managing event-condition-action rules in a database system, the method comprising the computer-implemented steps performed by said database system of:

storing, in a database managed by said database system, rule data that defines a composite event comprised of two or more primitive events, at least one condition related to the composite event, and at least one action to be performed upon satisfaction of said at least one condition;

detecting a first database event as an occurrence of a first one of the primitive events;

determining whether the first database event satisfies a first sub-condition of said at least one condition, wherein said rule data indicates that satisfaction of said first sub-condition is not sufficient to satisfy said at least one condition;

persistently storing in the database results data that indicates that said first sub-condition was satisfied by said first database event;

detecting a second database event as an occurrence of a second one of the primitive events;

reading said results data from said database;

determining whether the at least one condition is satisfied based on the results **data read from the database** and the second database event.

Nonetheless, the Office Action adds Hellerstein for the prospect that it teaches those highlighted claim limitations, through Hellerstein's system for processing historical event data.

At the outset, it is important to understand that Applicant does not only claim the notion of Event-Condition-Action processing. Nor does Applicant claim only the notion of storing event data in a database. Instead, what Applicant claims is a novel technique for incremental evaluation of conditions with respect to primitive events that comprise a composite event. Specifically, Applicant's claim 42 features (1) durably storing results data indicating that a first sub-condition of a condition related to a composite event is satisfied by a first database event, and then (2) determining whether the condition as a whole is satisfied based on a second detected database event and the durably stored results data. In contrast to conventional composite event detection mechanisms, the approach of Applicant's claims is not constrained by the amount of physical memory available to store a set of Event-Condition-Action rules, thereby facilitating processing of much larger sets of rules.

This is not the same as conventional composite event detection mechanisms, such as described in Chakravarthy and Etzion. In conventional composite event detection, there is a focus on speed in processing events against a rule set. These conventional mechanisms store data structures for processing events against rule sets in physical memory (e.g., volatile Random Access Memory (RAM) or main memory of a computer) where they can be accessed much more quickly than if stored on a larger but slower storage medium such as a hard disk. For example, Chakravarthy describes data structures such as an "event graph" and "operator trees" that are implemented in physical memory of a computer (see, e.g., Chakravarthy, p. 615 stating "The local composite event detector and the application share the same address space and our event detector uses an event graph similar to operator trees.")) In Etzion, in-physical-memory linked list data structure are used to detect composite event occurrences (see, e.g., Etzion, FIG. 3 illustrating a data structure into which the Etzion system maps event instances it receives). To support larger ECA rule sets than can be supported by mechanisms limited by the amount of physical memory, Claim 1 involves durably storing the results of such incremental evaluations in a database. The Office Action acknowledges that Chakravarthy and Etzion do not teach Applicant's technique for incremental evaluation, so the point need not be belabored.

However, the Office Action seems to believe that Applicant's technique can be recreated simply by bolting on Hellerstein's system for processing historical event data onto some combination of Chakravarthy's and Etzion's systems for composite event detection. However, Hellerstein's system does not provide enough teaching to convert a Chakravarhty-Etzion system into one that supports Applicant's technique for incremental evaluation of conditions with respect to primitive events that comprise a composite event. Importantly, Applicant's claim limitations are in terms of persistently storing results data, not event data.

For example, consider the following language of Applicant's claim 42:

persistently storing in the database **results data** that indicates that said first sub-condition was satisfied by said first database event;

detecting a second database event as an occurrence of a second one of the primitive events;

reading said **results data** from said database; and

determining whether the at least one condition is satisfied based on the **results data** read from the database and the second database event.

(Emphasis added.) As shown by the foregoing claim limitations, Applicant's invention of claim 42 is directed to (1) durably storing results data indicating that a first sub-condition of a condition related to a composite event is satisfied by a first database event, and then (2) determining whether the condition as a whole is satisfied based on a second detected database event and the durably stored results data.

That Hellerstein does not teach or suggest such a concept is made clear by the Hellerstein reference itself. For example, Hellerstein (Paragraph [0041]) states:

> In step 302, the event management decision support system reads previously accumulated **event data** into an event cache. The previously accumulated data is stored in memory associated with the event management decision support system, e.g., Event DB 180 in FIG. 1, prior to being read into the event cache. The previously accumulated event data represents **historical event data**. It is to be understood that the term "historical," as used herein, refers to **event data that was generated by network devices and received by the event management system at some prior time**. The time period from which the data is drawn may depend on the event management application. Thus, for example, the event data may be data generated and received between a point in time in the immediate past and some earlier relative point in time. Therefore, the historical event data accumulated over the desired time period is read from the Event DB into the event cache of the event management decision support system. It is this event data that is used to generate the one or more correlation rules.

(Emphasis added.) The event data stored in the Event DB described in Paragraph [0041] refers to data that was generated by network devices. The event data does not refer to results data that indicates that a sub-condition of a condition related to a composite event was satisfied. In other words, the event data represents an occurrence of an event, not whether an event satisfied a sub-condition of a condition related to a composite event. Consequently, the combination of Chakravarthy, Etzion, and Hellerstein asserted in the Office Action does not satisfy at least the following feature of claim 42 when taken as a whole:

> persistently storing in the database results data that indicates that said first sub-condition was satisfied by said first database event;

Moreover, Hellerstein describes an expert system for offline construction of correlation rules for event management; it does not describe techniques for runtime detection of composite event occurrences within a database system. The focus of Hellerstein is on the construction of correlation rules by analyzing historical event data. It involves a human analyzing historical event data or a computer executing data mining algorithms on historical event data stored in an event cache.

Applicant's claims, in contrast, focuses on runtime detection of composite event occurrences within a database system. Significantly, Applicant's claims provide a design that supports the processing of detected events against large rule sets – larger than those that can be supported by conventional composite event detection systems constrained by the amount of available physical memory. While Hellerstein's system may be helpful in construction of those rule sets, it does not provide any teaching about how to scale the processing of events against rule sets. Consequently, Applicant's approach of claim 42 for runtime evaluation of a rule set takes up where Hellerstein's approach for creating a rule set leaves off.

In the Advisory Action, the Examiner asserts that the claims do not distinguish between "results" data and "event" data. This is simply untrue. Claim 42 distinguishes between data that is capable of satisfying a condition (i.e., event data) and data that indicates whether a condition is satisfied by an instance of event data (i.e., results data). For example, Claim 42 recites "persistently storing in the database results **data that indicates that said first sub-condition was satisfied by said first database event**" (emphasis added). As shown by this limitation of Claim 42, Applicant's claims clearly distinguish between results data and event data. And while a combination of Chakravarthy, Etzion, and Hellerstein may teach or suggest persistently storing event data in a database it does not in any way teach or suggest "persistently storing in the database results data that indicates that said first sub-condition was satisfied by said first database event" as featured in Claim 42.

For the reasons stated above, it is respectfully submitted that Chakravarthy, Etzion, and Hellerstein, either individually or in a combination, do not teach or suggest all of the limitations of Applicant's claim 42. Accordingly, it is believed that Claim 42 distinguishes over the cited art and the Examiner's rejection of these claims under Section 103(a) should not be sustained. Applicant's other independent claim, Claim 51, recites similar limitations and is allowable over Chakravarthy, Etzion, and Hellerstein for the same reasons.

Respectfully submitted,
HICKMAN PALERMO TRUONG & BECKER LLP

Dated: July 9, 2009

/AdamCStone#60531/
Adam Stone
Reg. No. 60,531